International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Aug-2022)

https://dx.doi.org/10.12785/ijcds/120153

Protocol for Identity Management in Industrial IoT based on Hyperledger Indy

Cristina Regueiro[1], Iván Gutierrez-Agüero[2], Sergio Anguita[1], Santiago de Diego[1] and Oscar Lage[1]

[1]*TECNALIA, Basque Research and Technology Alliance (BRTA), Donostia-San Sebastián, Spain*

Received 5 Nov. 2021, Revised 10 May 2022, Accepted 15 Jul. 2022, Published 6 Aug. 2022

Abstract: This paper presents a protocol for Identity Management in Industrial IoT enabled devices, that is based on the principles of Self-Sovereign Identity. The Industry 4.0 transformation has led to the Industry sector digitalization and one of its major challenges is to uniquely identify the Industrial Internet of Things unattended devices. The digital identity management must allow increasing the security and control, and it has been evolving towards a model where the device acquires the responsibility for managing its own data through Self-Sovereign Identity. This paper studies why the Self-Sovereign Identity approach is suitable for the industrial IoT particularities, properly justifying its use. Furthermore, it analyzes the actors and roles involved in an industrial identity environment, and it addresses a protocol that defines how data should be exchanged over an Hyperledger Indy public permissioned Distributed Ledger Technology network as Sovrin. The paper applies the proposal to a reference use case, filling the gaps that are not currently specified in the literature for a successful Industrial Internet of Things identity management operation.

Keywords: Industrial Internet of Things; IIoT, Industry 4.0, Sovrin, Hyperledger Indy, Identity, Self-Sovereign Identity, SSI

1. Introduction

Industry is a sector that is constantly developing and giving rise to a series of revolutions that have transformed and improved its activity. The recent adoption of digital technology has led to the industry transformation called Industry 4.0, improving business operations and revenue growth by transforming products, supply chains and customer expectations [1].

In this context, the Industrial Internet of Things (IIoT) has been the latest innovation to improve productivity and industrial efficiency, intersecting information technologies (IT) and operational technologies (OT) [2]. OT refers to Industrial Control Systems (ICSs), including Supervisory Control And Data Acquisition (SCADA) systems, Distributed Control Systems (DCSs), Remote Terminal Units (RTU) and Programmable Logic Controllers (PLCs), among others. The convergence of IT and OT provides greater and safer system integration in terms of automation and optimization, as well as better visibility of the supply chain and logistics [3].

Industrial security is critical because the consequences of the lack of it could have devastating real-world effects [4]. Current hyperconnected industries are as reliable and secure as the technology serving them. While any data breach or a ransomware attack can cause significant losses in the IT, the impact is mainly financial or reputational. However, attacking to OT systems can introduce physical consequences, with effects that ripple throughout entire communities. Well-known incidents, such as Stuxnet [5], a malicious computer worm targeting SCADA systems, or the Mirai botnet [6], which infected several general-purpose IoT devices (IP cameras, routers, etc.), highlight the great importance of including a security vision when developing IIoT based solutions. Such an approach, known as ”security by design”, is recommended for any system and indispensable in the case of critical infrastructures, such as many industrial environments. Therefore, the security of IIoT devices is a crucial aspect [7].

One of the major challenges is to uniquely identify IIoT devices to facilitate transactions among entities, especially in an increasingly digitized world such as the industrial environment [8]. As the number of IoT devices in the industrial context is steadily increasing, the identification of IIoT devices becomes mandatory as, unlike a few years ago when everything was inside a local network protected by perimeter security elements, nowadays everything is worldwide hyperconnected. In this context, good identity management practices are needed for several security mechanisms, such as authentication, authorization, non-repudiation, secure exchanges, etc. Therefore, identity is essential to secure IIoT and avoid many attacks such as

E-mail address: cristina.regueiro@tecnalia.com, ivan.gutierrez@tecnalia.com, sergio.anguita@tecnalia.com, santiago.dediego@tecnalia.com, oscar.lage@tecnalia.com

those based on impersonation or identity theft (for example, the Sybil attack [9], which happens when a device simultaneously uses multiple identities).

The current model of implementing IIoT digital identities is insufficient [10]. There is a lack of a clear definition of how IoT should be identified or represented, resulting in non-interoperable identities and the creation of silos where devices can only interact with systems and devices that have a common manufacturer. As there is any legislative requirement for creating IIoT devices with an authenticated digital identity, current IIoT devices lack authentication and often reduce their protection against cybercriminals to the use of static credentials. Unless security becomes mandatory, manufacturers will continue reducing the price at the expense of security. However, it should be nowadays imperative that properly authenticated IIoT devices identity is in-built into devices at the point of manufacture and it should be updated in a continuous and automated process throughout the device lifecycle.

The present research work analyses why traditional identity management systems are not suitable for IIoT and proposes a detailed protocol for identity management in IIoT devices based on the novel Self-Sovereign Identity technology. The different sections of the paper are organized as follows: Section 2 introduces the traditional Identity Management Systems and identifies their limitations for IIoT. Section 3 presents Self-Sovereign Identity technology and justifies its applicability in the IIoT environment. Section 4 describes the proposed SSI-based identity management protocol considering the particularities of IIoT devices. Finally, some conclusions are drawn in Section 5.

2. Traditional Identity Management Systems

An Identity Management System (IdMS) refers to a system which manages identity information through a set of operations, including registration, updating, revocation and searching. An identity is often considered to be simply a set of access credentials, but in reality, it is, in addition to access credentials, a set of attributes or information related to a specific entity that represents it. Most IdMS currently involve three independent parties: users with their own identity, identity providers (IdP) in charge of identity management, and service providers, which rely on the IdP to verify the identity given by the user when providing a service.

Nowadays, there are mainly three different identity models [11] [12]:

- Centralized: It includes a service provider which acts as its own IdP. The main disadvantage is the large number of identities the user needs (one per service provider), which is especially relevant in the IIoT context due to the large number of services that a single IIoT could potentially need. In addition, the IdP is a clear Single Point of Failure (SPoF); if it fails the entire identity management system will fail. For that reason, centralized IdP are common targets for phishing attacks, as evidenced by the data breaches suffered by Equifax in 2017 [13] or Facebook in 2018 [14]. In addition, the proliferation of IdPs leads to fragmented identities spread across the Internet, greatly increasing the cost of suitable identification. Moreover, using a centralized IdMS scheme where the huge amount of existing IIoT identities should be maintained by a third-party means a huge performance decrement with the number of IIoT identities increment.

- Federated: It includes a central IdP allowing users to authenticate themselves with different service providers using the same identity. It refers to the "Single Sign On" mechanism in which a single instance of identity grants access to all service providers depending on the same IdP. OpenID Connect, SAML and OAuth are the three most used federated protocols [15], but they still have several security vulnerabilities [16]. Although usability and performance are improved, the IdP still suffers from SPoF and security and confidentiality (theft, loss of data, human faults) are not achieved due to the involvement of intermediate IdPs.

- User-centric: This model allows the user to completely control his identity. At the request of the service provider, the user can determine to what extend he wants to share or restrict exposure of his attributes; the IdP requests the user consent before sharing any identity attributes with the service providers. For example, Windows CardSpace [17], a tool for helping users to manage their identities in Windows, was discontinued due to several security vulnerabilities [18]. In general, the SPoF and security problems persist as the user attributes are still stored in an IdP.

Although IdMSs have evolved to increase the security and control that users have over their own identity, there is still a need for a third-party IdP which stores identity attributes meaning SPoF and confidentiality risks. This threat is especially critical in IIoT environments with unattended devices and with networks typically isolated from the Internet or lacking connectivity to other networks [19]. Besides, system availability and business continuity are always maximized in industrial environments; this is why identity schemes that constitute a SPoF are currently a risk for the industry. For this reason, IIoT requires a new identity management paradigm to address the confidentiality and security concerns of existing identity management models. In this context, Self-Sovereign Identity (SSI) appears as a suitable identity management model for the IIoT [20] [21]. It follows the idea behind the user-centric identity model of giving back the user full control over his identity and allowing users to reveal and share only the necessary attributes with any service provider; in other words, only users have the right to manage their own identities. In addition, it adds an extra security layer as, unlike other models, it

provides a direct trusted communication channel between users and the service providers without having to go through an intermediary IdP. By this way, several security risks are reduced: SPoF disappears and confidentiality is maximised. However, it should be noted that the user still needs to have previously obtained his identity attributes from the IdP in order to be able to present them to any service provider.

3. Self-Sovereign Identity

A. Principles

SSI is based on 10 principles to ensure the user control over his own identity [22]. Although it was originally thought for the identity of individuals, these principles should also be covered in the IIoT context.

- Existence: Users must have an independent existence. SSI is based on the "I" as it only refers to attributes of the "I" that already exist. In the context of IIoT, the identity refers only to IIoT devices from the point of manufacture, allowing them to exist.
- Control: Users must control their identities as they are the ultimate authorities on their identities. They should always be able to refer, update or even hide it. IIoT devices, which increasingly include higher levels of intelligence, should control their own attributes, increasing security and confidentiality.
- Access: Users must have access to their own attributes, with no hidden data and no gatekeepers. IIoT devices should also be able to access all their own attributes when desired.
- Transparency: Systems and algorithms must be transparent; anyone should be able to examine how they work.
- Persistence: Identities must be long-lived. Nowadays, identities should last until they are outdated. In addition, user's identity should be modified or removed as appropriate over time. This fact is important for IIoT devices as many of their features may be degraded over time and should no longer be part of the IIoT identity.
- Portability: Any identity that manages information and services shall be able to use the mechanisms that best suit its needs without losing any capability provided by the identity. The IIoT devices shall be allowed to change the application that manages the identity at any time, and the identity information and purposes must remain the same.
- Interoperability: Identities should be as widely usable as possible. Ideally, they should cross international boundaries creating global identities, without losing user control. As in the previous case, this is essential for IIoT, as devices are manufactured in an organisation but will be used in a different one; their identity should be persistent among different organisations, different countries, etc.
- Consent: Users must agree to the use of their identity; there should be a previous "consent".
- Minimisation: Disclosure of claims must be minimised. Only required attributes should be shared, increasing confidentiality as best as possible.
- Protection: The rights of users must be protected.

B. Architecture and components

SSI is a digital identification scheme in which the subjects whose identity is created acquire responsibility for managing how, when and with whom they share their personal data [23]. To this end, it allows the creation of "digital identity proofs" (presentations) based on his own identity attributes.

In general, there are three actors involved in the SSI schema:

- Issuer, provides verifiable credentials with identity attributes related to the user. It creates and signs credentials.
- Holder, locally stores and controls the credentials about himself. In the IIoT context, it will be the IIoT device.
- Verifier, needs to identify a user's attribute or a set of them based on verifiable credentials by trusted issuers. Verifier does not need to store any user (holder) data, but only needs to verify it. This verification is based on validating the holder's provided credentials proof, in where requested claims by the verifier are attested. Depending on the verifier requirements this attestation could disclose credential claims values or be a private attestation based on Zero Knowledge Proofs (ZKPs).

It is important to highlight the need for the holder to have computing (for providing verifiable proofs from its identity credentials), storage (for storing identity verifiable credentials) and connectivity (for interaction with other actors) capabilities. In the IIoT context, it is conceivable that not all IIoT devices meet these requirements. Therefore, although the holder is in general the actor who manages his own identity, in case the necessary capacities are not covered by the IIoT device, the role of the holder is divided into two actors: subject, who is the actor the identity attributes are defined about, and the holder, who is the one who controls and manages the subject's identity (in the IIoT scenario, this could be, for example, the IIoT device owner). In this case, the subject is a passive role who does not perform any operation.

Regarding the system components involved in an SSI solution:

- Credential, it is a digital certificate containing identity attributes of the holder it is associated with. It is issued by the issuer.
- Wallet, it is a secure credential storage system used by the holder. It may be local (or not), but the holder must have control over the credentials inside it. It is also needed by all the actors for cryptography functions execution.
- Presentation, it is a digital evidence shared by the holder with the verifier to prove certain characteristics of the subject's identity based on the received credentials.

The verifier is not necessarily related to the issuer, so, the only way to digitally prove that credentials have been really issued by a trusted issuer, and have not been modified in any way, is by means of digital signatures. Digital signatures are based on applying a private key in a digital signature algorithm over a specific information. The signature can be verified using the same digital signature algorithm but with the associated public key. In the SSI context, digital signing is at least applied by the issuer to the credentials, becoming verifiable credentials, and by the holder to the presentations, becoming verifiable presentations. Public keys associated to issuers and holders should then be known. They could be saved in a centralized database, but there may be integrity issues due to a third-party being able to modify the public keys, availability issues (SPoF), due to probably system bringing downs and in addition the operator of this centralized database would have visibility of all relationships between different subjects.

For these reasons, instead of a centralized database, Distributed Ledger Technology (DLT) can act as a suitable global repository for public key identifiers in SSI, as it solves several problems from traditional databases [24] [25]:

- Trust: DLT is based on a decentralized network of computers which is not owned by one single party, so it is not necessary to trust on any specific party.
- Integrity: Immutability is an inherent property of DLT guaranteeing tamper-proofed data.
- Availability: DLT is a network of computers across the globe and bringing down it is near to impossible.

DLT creates globally distributed databases that can serve as a source of truth for public keys without being subject to SPoF. This is the reason why DLT generally fits into the SSI infrastructure for registering and resolving public keys. The differences between considering different types of DLT (permissioned, permissionless, hybrid, etc.) do not have a real impact on the identity management system. In this sense, SSI uses Decentralized Identifiers (DID) as a unique and global identifier of every person or object involved in the process.

Each DID is associated to a DID Document, describing its properties, such as the associated public key (and additional public keys that are authorized to perform actions in its name) or service endpoints for interacting with the specific DID. It is necessary for the signing process in verifiable credentials by the issuer and in verifiable presentations by the holder; and for identifying the subject of a verifiable credential. This architecture eliminates unnecessary third-party identity providers and highly reduces security risks.

Summarizing, the basic architecture of the DLT-based SSI solution in the IIoT scenario is shown in Figure 1.

C. Current Status of the technology

Several SSI frameworks have also been developed for identity management. Sovrin [26], Serto [27] and Civic [28] are worthy of attention projects. Among these options, Civic presents portability issues for IIoT devices, highly limiting their interoperability capacity while Serto lacks in terms of security and privacy [29]. Sovrin fails in the usability-related aspects as its technical internal operation is quite complex [29]. However, usability is not so relevant in the IIoT context since IIoT devices will execute automatic applications for interacting with the identity management system, turning Sovrin into the most suitable SSI identity management system for IIoT. The same conclusion is also obtained in [30], where a very precise comparative analysis is performed, highlighting Sovrin advantages in terms of privacy and security.

The projects were initially focused on people's identity; however, they are a good starting point to be extended to IoT devices. The use of DIDs and verifiable credentials for IoT has been proven to be more suitable than other methods, such as X.509 certificates [20]. Furthermore, potential use cases for SSI in the industrial environment have been identified [31] [32]; for example, using SSI for tracing the origin of IoT devices [33] or for providing IoT-as-a-service solutions [34]. These studies tend towards SSI but without giving a proper justification as to why SSI better fits IoT scenarios. Additionally, the existing literature mainly describes the "big picture" but without focusing on technical details such as the specific verifiable credentials exchange protocol involving IoT devices which is deeply defined in this research work.

Moreover, SSI is nowadays gaining a lot of support at a governmental level, being part of some of the more important worldwide security initiatives, such as that in Europe (EBSI, European Blockchain Services Infrastructure) [35], Canada (PCTF, Pan-Canadian Trust Framework) [36] or China (Distributed Identity Alliance, DIA) [37].

D. Applicability in IIoT

Current IIoT devices have been shown to have several security vulnerabilities such as, weak, guessable, or hardcoded passwords, lack of secure update mechanisms

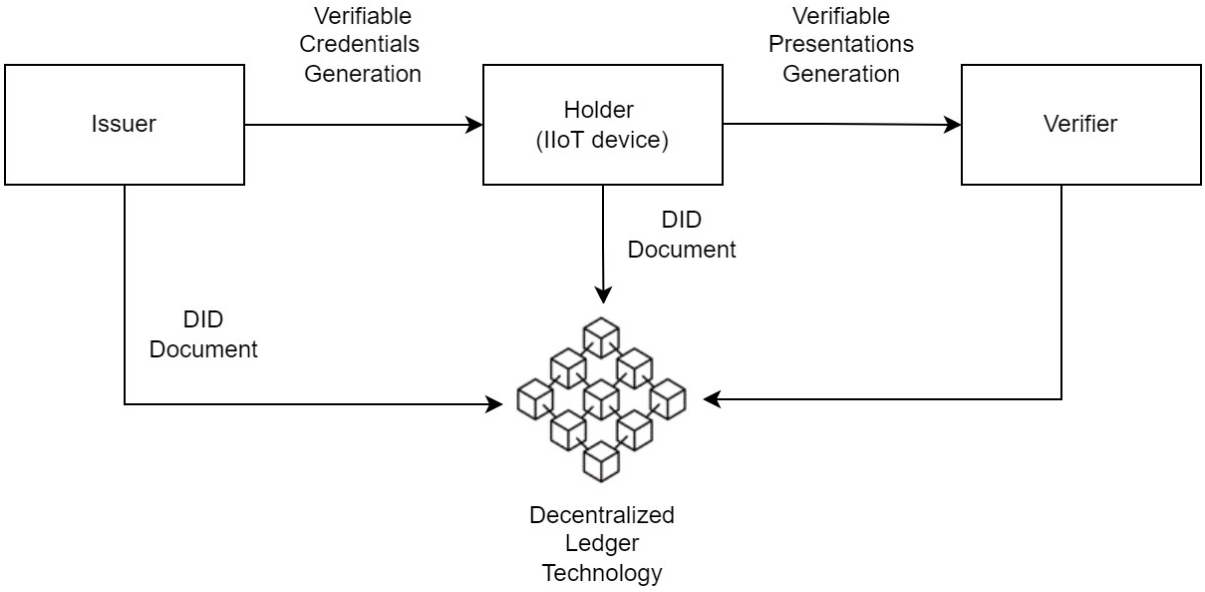

Figure 1. DLT-based SSI Global Architecture

or insufficient confidentiality protection [38]. The problem is not only the lack of a clear cybersecurity vision; manufacturers do not invest very often in cybersecurity measures because it means an extra cost for each product, reducing their competitiveness. Fortunately, this situation has recently started to change with manufacturers collaborating with researchers and cloud service providers to design new security measures and protocols for IoT devices [39]. However, there is still a lot of work to be done [40]. IIoT devices have several characteristics that directly affect their identity management [7] turning the traditional IdMSs into inadequate and creating a need for SSI based systems for identity management:

- Immensity and scalability: considering the large number of IIoT devices, centralized identity management systems are very unrealistic, at least in terms of scalability. Federated approaches aim to solve this problem, but they still maintain limitations that are solved with decentralized identity schemes.
- Interoperability: the huge diversity and heterogeneity of IIoT devices implies interoperability problems (especially between different manufacturers), making the definition of a global identity management system for IIoT devices very difficult. Standards, such as those considered by current SSI technologies, are consequently needed.
- Mobility and ubiquity: Many of the current IIoT devices are mobile devices. For that reason, the IIoT identity management system should be capable of providing authentication and authorization with independence of the devices location. SSI based solutions in which each IIoT devices stores and handles its own identity attributes perfectly fits with mobile devices.
- Privacy: IIoT devices usually handle and store sensitive information that should not be shared unnecessary. For that reason, intermediaries should preferably be avoided, as happens in decentralized solutions.
- No supervision: IIoT devices are usually unsupervised; their functionality is usually performed by automatic tools. This feature also fits with SSI concept as related actions can be also automatized.
- Limited features: Many of current IIoT devices have limited features; they are typically equipped with 16-bit or even 8-bit microcontrollers [41] and have limited RAM and disk space. In terms of connectivity, they are usually connected to the Internet via Ethernet or low-quality WiFi connections. However, as it is shown in [42], most IoT devices can successfully implement DIDs and verifiable credentials and presentations, being consequently able to support SSI solutions. However, proxy-based approaches or guardianship concepts where a physical person oversees managing the IIoT device identity [43] are suitable for extremely constrained devices.

4. SSI Operation in the IIoT Context

According to the previous section, the identity management in the IIoT context should be based on Sovrin. Sovrin is a public identity network (everyone can use it) built on the public permissioned Hyperledger Indy DLT technology.

Hyperledger Indy describes a generic protocol for identity management in the “Indy Story Walkthrough” [44]. However, this protocol was created for dealing with people’s identity and should be updated according to IIoT devices

particularities. Although the applicability of SSI to the IIoT context has been extensively proven [20], [31], there is still a lack of specific studies about how to adapt it the particularities of IIoT devices.

A. Actors

A reference use case, shown in Figure 2, has been defined for showing how the identity management of industrial machines is carried out following the SSI schema. The idea is to build and manage an industrial machine identity based on specific attributes (productive capabilities) certified by its manufacturer as verifiable credentials that can then be proved to a potential customer in a secure way.

In Hyperledger Indy, all three roles involved in the reference use case will need an agent to securely exchange information between them. The agent requires access to a digital wallet in order to perform cryptographic operations and save identity key pairs and other private data such as DID, credential definitions or revocation data.

- The manufacturer, as an issuer, will perform its issuing functionality by means of an automated script. This script will implement the agent with the verifiable credentials' issuance functionality as well as the wallet and secure exchanging functionalities.
- The machine, as a holder, will perform its functionality by means of an automated script since it is an unsupervised device. In this case, in addition to the wallet and secure exchanging functionalities, this agent will also hold and process the verifiable credentials received from the issuers as well as provide verifiable presentations.

  The considered industrial machine fulfils the computing, storage and connectivity requirements to act as a complete holder, being able to manage its own identity. Anyway, as it has been mentioned before, even if the machine does not fulfill the necessary requirements (it would be just the subject) it needs an additional actor to manage its identity (holder), the subject is still a passive element with no influence on the SSI operation and it is just the holder the one who operates on behalf of the subject.
- Customers, as verifiers, need to verify specific attributes about the specific machines they are going to use/buy. Ideally, in a future where the SSI concept is widespread, each customer organization would have its own SSI-based verification application deployed at their own premises. However, as nowadays SSI is not widely extended and there is not a unique customer (different customers from different companies are expected), the most suitable option for the verification functionality is by means of a web application with a user/password-based authentication process in order to be directly available for worldwide customers. This agent will perform wallet and secure exchange functionalities as well as verifiable presentations validation. This scenario fits with Indy proposed cloud agent implementation for cloud-based interactions.

  The web application will be part of a "machine identity management" web service provided by the manufacturer for its customers. This web service will also include the storage of information related to the produced machines as well as the provision of web lockers for secure information exchanges with the machine.

B. Operation Protocol

As it has been introduced before, the identity management process follows the protocol described in "Indy Story Walkthrough" [44]. However, there are some open issues that have been identified and particularized for the IIoT context, especially those related to the way in which the information between the involved actors is exchanged.

The proposed protocol considers the use of web lockers for the information exchanging. Web lockers are secure web-based storage services for information exchanges between known (manufacturer and machines) or unknown actors (machines and customers). In order to securely share the web lockers among different actors (different machines and customers), confidentiality must be guaranteed. For this purpose, end-to-end encryption is suggested; the proposed protocol makes heavy use of Indy SDK provided crypto functions (such as cryptoAnonCrypt, cryptoAnonDecrypt, packMessage and unpackMessage). In this way, any interaction that involves sensitive data from and to the web lockers is protected and sealed. The encryption algorithm will only allow to message recipients to decrypt the message. Those recipients need to be configured and setup prior message sending to the web locker. In this way, the proposed protocol becomes secure and safe against eavesdropper and Man-in-the-Middle (MitM) attacks.

The process, shown in Figure 3, starts with the onboarding process for every involved actor, creating their associated DID and recording them in their own wallets. The following steps are as follows:

- Step 1: The manufacturer starts its operation by creating and registering a credential schema in the DLT. The credential schema is the semantic structure that describes the list of attributes for a specific credential. This schema will define the relevant machine parameters/attributes such as, machine ID, fabrication date, warranty end date, energy consumption etc. The specific attributes to be included in the schema should be defined in advance (variable parameters). It is not possible to update an existing schema; in case an update is needed, a new credential schema with the updated attributes must be created and specifying a new credential schema version value or name.

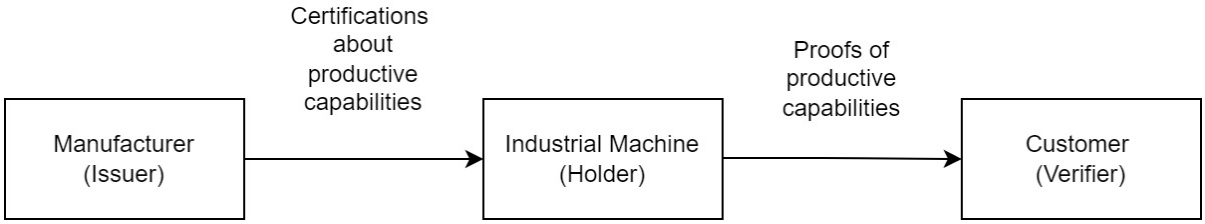

Figure 2. IIoT reference use case

- Step 2: The manufacturer creates and registers a credential definition in the DLT based on the credential schema defined and registered in Step 1. The credential definition announces who will issue verifiable credentials for a specific credential schema, what type of signature will be used, how revocation will be handled, etc. As in the previous case, it is not possible to update a credential definition; a new one should be created if needed.

- Step 3: Although credentials can be issued with an expiration date, they should be able to be revoked if needed (error issuance, unexpected damage, etc.). For this purpose, the manufacturer creates and registers a revocation registry in the DLT, referencing the credential definition from Step 2 and the revocation handling way. Every time a credential is revoked, this register is accordingly updated.

- Step 4: The manufacturer generates a credential offer for the machines it manufactures indicating the associated credential schema (attributes). This credential offer will be encrypted and stored in a "credential offers web locker" within the manufacturer web server until the any machine accesses it. The "credential offers web locker" URL will be known by all the machines produced by the manufacturer (variable parameter for the automated script executed in the machine).

- Step 5: A customer, who wants to verify certain attributes of the machine, will access the web application for verifiable presentations validation from a browser in their own computers/devices. User/password-based authentication will be requested.

- Step 6: The customer creates a verifiable presentation request by means of the web application, indicating the attributes associated to the machine he wants to validate. This verifiable presentation request is encrypted to be securely saved in a "verifiable presentation requests web locker" within the manufacturer web server until the referenced machine accesses it. The "verifiable presentation requests web locker" URL will be known by all the machines produced by the manufacturer (variable parameter for the automated script executed in the machine). The customer will define a timeout for waiting for the required verifiable presentation (variable parameter for the automated script executed in the machine). If timeout happens, the verification application will automatically finish this process.

- Step 7: The machine periodically checks the "verifiable presentation requests web locker" to check if there are new verifiable presentation requests. The specific period for this checking process can be configured for each machine (variable parameter for the automated script executed in the machine).

- Step 8: In the case of a new verifiable presentation request, the machine will take and decrypt it from the "verifiable presentation requests web locker"; it will then identify the required attributes and predicates that has been requested to prove. These attributes can be proved based on the verifiable credentials that may already be available in the machine from previous processes. However, in case the stored verifiable credentials do not prove the required attributes, new verifiable credentials will be needed.

- Step 9: As new verifiable credentials are required; the machine checks the "credential offers web locker" to check if there are new credential offers. The "credential offers web locker" is only checked when new verifiable credentials are needed; this process will be periodically repeated since then (variable parameter for the automated script executed in the machine) until the required offer is shown or a time-out happens (variable parameter for the automated script executed in the machine). If time-out happens, the machine will automatically finish this process, notifying the manufacturer (by pop-up, email..., this is a variable parameter) about the attributes required by the customers for future new schemas definition if applicable.

- Step 10: In the case of a new credential offer, the machine will take and decrypt it from the "credential offers web locker". The machine will then analyze the credential schema and credential definition from the taken credential offers, looking at the details stored in the DLT in order to determine if the required attributes can be proved with a new verifiable credential following the obtained credential offers.

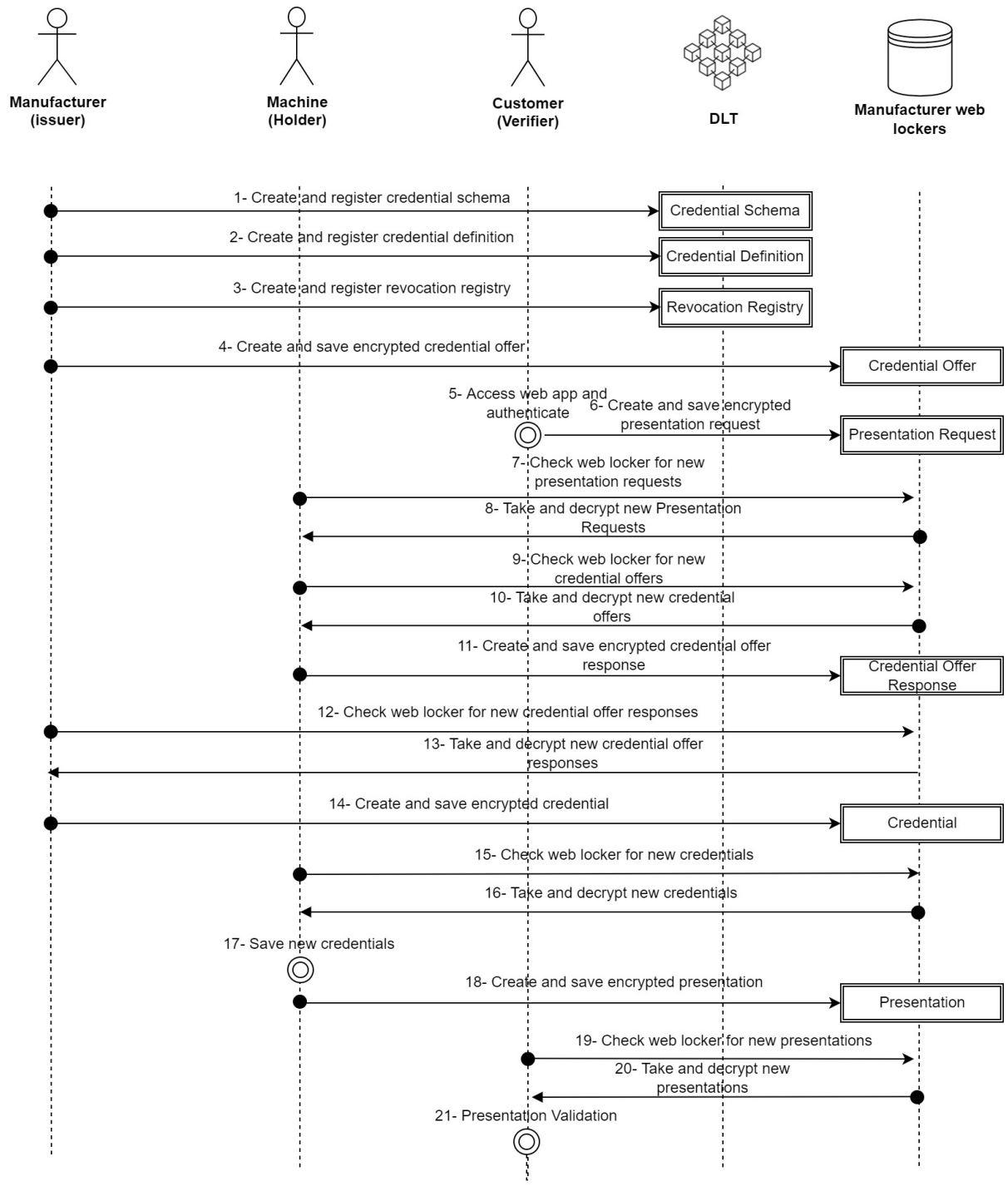

Figure 3. DLT-based SSI Operation for IIoT

- Step 11: In case the attributes from a credential offer match the attributes to be proved, the machine will make and encrypt a credential offer response to the manufacturer, asking him to fill in the verifiable credential attributes following the details from the credential offer. The credential offer response will be stored in a "credential offer responses web locker" within the manufacturer web server until the manufacturer accesses it. The "credential offer responses web locker" URL will be known by all the machines produced by the manufacturer (variable parameter for the automated script executed in the machine).
- Step 12: The manufacturer periodically checks the "credential offer responses web locker" to check if there are new credential offer responses. The specific period for this checking process can be configured for the manufacturer (variable parameter for the desktop application).
- Step 13: In the case of a new credential offer response, the manufacturer will take and decrypt it from the "credential offer responses web locker".
- Step 14: Once the credential offer response is received the manufacturer, it will fill in and sign the specific values for each of the schema attributes for a particular machine, generating the verifiable credential. This verifiable credential will be then encrypted and stored in the "verifiable credentials web locker" within the manufacturer web server until the referenced machine accesses it. The "verifiable credentials web locker" URL will be known by all the machines produced by the manufacturer (variable parameter for the automated script executed in the machine).
- Step 15: As the machine is waiting for a new verifiable credential, it checks the "verifiable credentials web locker" to check if there are new verifiable credentials. In the case of a new verifiable credential, the machine will take it and delete it from the credential locker. The "verifiable credentials web locker" is only checked when waiting for new verifiable credentials; this process will be periodically repeated since then (variable parameter for the automated script executed in the machine) until the required verifiable credential is shown or a timeout happens (variable parameter for the automated script executed in the machine). If timeout happens, the machine will automatically finish this process.
- Step 16: In the case of a new verifiable credential, the machine will take and decrypt it from the credential locker.
- Step 17: The machine locally stores the taken verifiable credential in its own wallet.
- Step 18: Following the customer verifiable presentation request from Step 7, as the machine now has the required verifiable credential to prove the requested attributes, it can create the appropriate verifiable presentation. This verifiable presentation will be then encrypted and stored in the "verifiable presentations web locker" within the manufacturer web server until the customer accesses it. The "verifiable presentations web locker" URL will be known by all the machines produced by the manufacturer (variable parameter for the automated script executed in the machine). The customer is notified of the existence of the requested verifiable presentation for demonstrating the required attributes (pop-up, email..., this is a variable parameter for every customer).
- Step 19: The customer will then check the "verifiable presentations web locker" to obtain the required verifiable presentation.
- Step 20: The customer takes and decrypts the new verifiable presentation and delete it from the "verifiable presentations web locker".
- Step 21: The customer will identify the requested attributes in the verifiable presentation checking its validity by verification in the DLT (signatures, revocation status...).

As a result, the customer can be sure that the attributes the machine is supposed to have are currently reliable and have been certified by the manufacturer himself, whom he trusts.

C. *Evaluation*

The proposed SSI-based IIoT Identity Management solution allows a secure characterization of industrial assets through verifiable credentials issued by trusted issuers (manufacturers, certification entities, etc.). Technically, Hyperledger Indy has already been proven to provide strong security guarantees due to the use of encrypted peer-to-peer connections for the credentials exchange as well as the use of unique identifiers for each relationship between actors [45]. It is also backboned by trustworthy Blockchain technology for the DID registration, which is considered secure by design, providing integrity, availability and traceability [46].Furthermore, the proposed protocol is based on the use of cryptographically secure web lockers servers for identity information exchange. Taking this into account, the level of trust of the proposed solution mainly depends on the level of trust in the issuer as verifiable credentials will be considered reliable if they are issued by a reliable issuer. For example, the manufacturer may be considered as a trustworthy actor to prove details about the manufacturing process of a specific machine (date, performance...). However, it will not be reliable for a certification process of certain standards compliance since only certification agencies are considered reliable for issuing compliance certificates.

Due to their high security level, verifiable credentials

could be used by IIoT devices for different purposes, such as authentication [47], access control policies application [48] and certification processes [49], highly improving the level of trust of these processes and facilitating their secure automation. So, this new identity management mechanism will improve other security mechanisms needed by IIoT devices.

Nowadays, most of the current IoT devices have already been demonstrated to be able to work with DIDs [42] as well as with Hyperledger Indy based verifiable credentials [34] with good performance results in terms of performance (CPU and RAM consumption). However, as explained before, some computing, storage and connectivity requirements need to be fulfilled by the IIoT device to be able to deal with its own identity and avoid needing guardianship solutions [43] which limit, to some extent, the idea behind self-sovereign identity.

5. Conclusions

This paper studies the suitability of following the philosophy of Self-Sovereign Identity for the industrial IoT environment, deeply analyzing the IIoT features. The paper also presents an SSI-based protocol for Identity Management in the IIoT context, particularizing it for IIoT devices and filling the gaps that are not currently specified. The proposed protocol solves how data should be exchanged over an Hyperledger Indy public permissioned DLT network, identifying and explaining the need of using web locker servers as a secure way for exchanging identity information. The actors and roles involved in the industrial identity environment have been identified and specific implementations features have been proposed based on the IIoT devices particularities. The result is a successful and secure identity management operation in the industrial domain.

Acknowledgment

This work has been supported by the Basque Country Government under the ELKARTEK program, project TRUSTIND (KK-2020/00054).

References

[1] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, "The expected contribution of industry 4.0 technologies for industrial performance," *International Journal of production economics*, vol. 204, pp. 383–394, 2018.

[2] V. Tsiatsis, S. Karnouskos, J. Holler, D. Boyle, and C. Mulligan, *Internet of Things: technologies and applications for a new age of intelligence*. Academic Press, 2018.

[3] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (iiot): An analysis framework," *Computers in industry*, vol. 101, pp. 1–12, 2018.

[4] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2015, pp. 1–6.

[5] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[6] G. Kambourakis, C. Kolias, and A. Stavrou, "The mirai botnet and the iot zombie armies," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 2017, pp. 267–272.

[7] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet of things Journal*, vol. 6, no. 2, pp. 1606–1616, 2018.

[8] A. Rajan, J. Jithish, and S. Sankaran, "Sybil attack in iot: Modelling and defenses," in *2017 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2017, pp. 2323–2327.

[9] J. Sengupta, S. Ruj, and S. D. Bit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, vol. 149, p. 102481, 2020.

[10] C. S. Writer, "The insecure iot device free-for-all needs to be urgently tackled," TECHMonitor, 2020.

[11] M. Sheng, Y. Qin, L. Yao, and B. Benatallah, *Managing the web of things: linking the real world to the web*. Morgan Kaufmann, 2017.

[12] L. Spinaci, "Digital identity's models," CollaboGate, 2018.

[13] P. Wang and C. Johnson, "Cybersecurity incident handling: a case study of the equifax data breach." *Issues in Information Systems*, vol. 19, no. 3, 2018.

[14] S. F. Mike Isaac, "Facebook security breach exposes accounts of 50 million users," The New York Times, 2018.

[15] S. Koussa, "Federated identities: Openid vs saml vs oauth," *WWW: http://www. softwaresecured. com/2013/07/16/federated-identities-openid-vs-saml-vs-oauth/(cit. 25. 04. 2016)*, 2013.

[16] S. Simpson and T. Groß, "A survey of security analysis in federated identity management," in *IFIP International Summer School on Privacy and Identity Management*. Springer, 2016, pp. 231–247.

[17] V. Bertocci, G. Serack, and C. Baker, *Understanding windows cardspace: an introduction to the concepts and challenges of digital identities*. Pearson Education, 2007.

[18] S. Gajek, J. Schwenk, M. Steiner, and C. Xuan, "Risks of the cardspace protocol," in *International Conference on Information Security*. Springer, 2009, pp. 278–293.

[19] X. Zhu and Y. Badr, "Identity management systems for the internet of things: a survey towards blockchain solutions," *Sensors*, vol. 18, no. 12, p. 4215, 2018.

[20] G. Fedrecheski, J. M. Rabaey, L. C. Costa, P. C. C. Ccori, W. T. Pereira, and M. K. Zuffo, "Self-sovereign identity for iot environments: a perspective," in *2020 Global Internet of Things Summit (GIoTS)*. IEEE, 2020, pp. 1–6.

[21] N. Kulabukhova, A. Ivashchenko, I. Tipikin, and I. Minin, "Self-sovereign identity for iot devices," in *International Conference on Computational Science and Its Applications*. Springer, 2019, pp. 472–484.

[22] C. Allen, "The path to self-sovereign identities," *Retrieved October*, vol. 31, p. 2017, 2016.

[23] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, 2016.

[24] D. Baars, "Towards self-sovereign identity using blockchain technology," Master's thesis, University of Twente, 2016.

[25] D. Van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-sovereign identity solutions: The necessity of blockchain technology," *arXiv preprint arXiv:1904.12816*, 2019.

[26] D. Khovratovich and J. Law, "Sovrin: digital identities in the blockchain era," *Github Commit by jasonalaw October*, vol. 17, pp. 38–99, 2017.

[27] A.-E. Panait, R. F. Olimid, and A. Stefanescu, "Analysis of uport open, an identity management blockchain-based solution," in *International Conference on Trust and Privacy in Digital Business*. Springer, 2020, pp. 3–13.

[28] civic, "Identity verification by civic," 2022.

[29] A. Satybaldy, M. Nowostawski, and J. Ellingsen, "Self-sovereign identity systems," in *IFIP International Summer School on Privacy and Identity Management*. Springer, 2019, pp. 447–461.

[30] N. Naik and P. Jenkins, "Self-sovereign identity specifications: Govern your identity through your digital wallet using blockchain technology," in *2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2020, pp. 90–95.

[31] P. C. Bartolomeu, E. Vieira, S. M. Hosseini, and J. Ferreira, "Self-sovereign identity: Use-cases, technologies, and challenges for industrial iot," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 1173–1180.

[32] P. N. Mahalle, G. Shinde, and P. M. Shafi, "Rethinking decentralised identifiers and verifiable credentials for the internet of things," in *Internet of things, smart computing and technology: A roadmap ahead*. Springer, 2020, pp. 361–374.

[33] T. Weingaertner and O. Camenzind, "Identity of things: Applying concepts from self sovereign identity to iot devices," *The Journal of The British Blockchain Association*, p. 21244, 2021.

[34] S. De Diego, C. Regueiro, and G. Maciá-Fernández, "Enabling identity for the iot-as-a-service business model," *IEEE Access*, vol. 9, pp. 159 965–159 975, 2021.

[35] ebsi, "European blockchain services infrastructure," 2022.

[36] D. Committee, "Pan-canadian trust framework overview," 2016.

[37] Plato, "Major chinese tech companies come together to announce the distributed identity alliance," 2020.

[38] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, 2015, pp. 1–7.

[39] C. H. Kim, T. Kim, H. Choi, Z. Gu, B. Lee, X. Zhang, and D. Xu, "Securing real-time microcontroller systems through customized memory view switching." in *NDSS*, 2018.

[40] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," *Future generation computer systems*, vol. 82, pp. 395–411, 2018.

[41] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of resource constrained devices in the internet of things," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 144–149, 2012.

[42] Y. Kortesniemi, D. Lagutin, T. Elo, and N. Fotiou, "Improving the privacy of iot with decentralised identifiers (dids)," *Journal of Computer Networks and Communications*, vol. 2019, 2019.

[43] S. G. T. Force, "On guardianship in self-sovereign identity," 2019.

[44] S. G. T. Force, "Indy walkthrough. a developer guide for building indy clients using libindy," 2018.

[45] H. Indy, "Hyperledger indy: Security and privacy," 2020.

[46] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[47] Z. A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 2020, pp. 71–78.

[48] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, and S. Guerreiro, "Ssibac: self-sovereign identity based access control," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 1935–1943.

[49] I. Barclay, S. Radha, A. Preece, I. Taylor, and J. Nabrzyski, "Certifying provenance of scientific datasets with self-sovereign identity and verifiable credentials," *arXiv preprint arXiv:2004.02796*, 2020.

Cristina Regueiro Cristina Regueiro. Telecommunications Engineer (2010) with a PhD in Information Technology and Communications in Mobile Networks (2017) from the University of the Basque Country. She started as a Researcher at the University of the Basque Country on issues related to digital signal processing, wireless communications, and mobile communications (2011-2017). She continued her research work at the Innovalia Association (2017-2018) with a deep focus on industrial communications, Industry 4.0, IoT, fog, and cloud computing. Her cybersecurity experience was later completed at Ikerlan (2018-2019) taking part in the cybersecure IoT and cybersecurity on digital platforms teams. In December 2019, she joined the cybersecurity and blockchain team within Tecnalia, where she is currently a senior researcher with a focus on cybersecurity solutions and blockchain-based systems.

Iván Gutierrez Iván Gutierrez-Agüero received his degree in Computer Science Engineering at UPV/EHU. Expert in Cryptography and Cybersecurity, he owns a master's degree in Software Engineering and Intelligent Systems. Iván actively contributes to the technology dissemination as ESIC Business and Marketing School Professor, as part of the Hyperledger Speakers Bureau and through specialized congresses and learning programs. He is also a peer reviewer for IEEE and SNCS. After working on neural network sets, ontological driven expert systems and automated image processing, he has grown as a software engineer with technical skills at cybersecurity and software design. As a member of Tecnalia, Iván has been working on research projects related to PKI, IdM, IAM, chip-based cryptography, contactless technologies, mobile devices and Blockchain.

Sergio Anguita Sergio Anguita is a MSc Computer Engineering from University of Deusto. From the very beginning, he has worked at DeustoTech Computing on s3lab area, focused on cybersecurity and privacy. He participated in multiple projects for different companies, related to web infrastructures, secure implementation of NFC solutions and massive data analysis for anomaly detection. Sergio also researched in mobile cybersecurity field, automating the process of reversing Android applications looking for malware behavior, threats, or potential risks for user's privacy. Sergio also leads an open source cyber security related project in which he previously researched, and now, he is part of the Tecnalia Cyber Security Research Team, working on the research of Blockchain solutions for industrial systems and processes.

Santiago de Diego Santiago de Diego Mathematician and IT Engineer from the University of Granada (UGR) and the master's degree in information security at the International University of La Rioja (UNIR). He currently works as a cybersecurity researcher at TECNALIA Research and Innovation and is pursuing a Ph.D. degree at the University of Granada under the supervision of Dr. Gabriel Maciá and Dra. Cristina Regueiro. His current research interests include distributed ledger technologies, cybersecurity for critical infrastructures, and identity management systems.

Oscar Lage Oscar Lage is Senior Researcher and Project Manager at TECNALIA, Security and Mobile researcher since 2003, he has taken part in several European and Spanish research projects like Internet of Energy, Chiron, nSHIELD, Safecity or Smart Urban Spaces developing innovative Information Security technologies based on Cryptographic algorithms. His current research interests include Identity and Access Management, Mobile Security and Mobile Payments. He develops and applies research mainly in eBanking and Ticketing, Wearable technologies, Ambient Assisted Living, eHealth and Heterogeneus Telecom Networks.